Attorney Docket 30695.21USU1

Patent Application

for

REDUCING INFORMATION TRANSMISSION TIME BY ADAPTING INFORMATION DELIVERY TO THE SPEED OF A GIVEN NETWORK CONNECTION

Inventor(s):

PAUL F. KLEIN

Prepared By:

Jason S. Feldmar
Gates & Cooper LLP
Howard Hughes Center
Suite 1050
6701 Center Drive West
Los Angeles, California 90045

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims the benefit under 35 U.S.C. Section 119(e) of the following co-pending U.S. provisional patent application, which is incorporated by reference herein:

Provisional Application Serial No. 60/214,281, filed June 22, 2000, by Paul F. Klein, entitled "REDUCING INTERNET OBJECT TRANSMISSION TIME BY ADAPTING INTERNET OBJECT DELIVERY TO THE SPEED OF A GIVEN INTERNET CONNECTION," attorney's docket number 30695.21-US-P1.

This application is a continuation-in-part of the following co-pending and commonly assigned patent applications, which applications are incorporated by reference herein:

United States Patent Application Serial No 09/711,660, filed November 13, 2000 entitled "METHOD AND APPARATUS FOR DETERMINING A RESPONSE TIME FOR A SEGMENT IN A CLIENT/SERVER COMPUTING ENVIRONMENT", by Paul F. Klein et al., attorney's docket number 30695.19-US-U1, which application claims the benefit of Provisional Application Serial No. 60/172,026, filed December 23, 1999, by Paul F. Klein, entitled "MEASURING RESPONSE TIME FOR VARIOUS SEGMENTS OF A STANDARD CLIENT/SERVER COMPUTING ENVIRONMENT BY DIRECTLY MEASURING ONE SEGMENT AND STATISTICALLY DERIVING RESPONSE TIME FOR THE REST," attorney's docket number 30695.19-US-P1;

United States Patent Application Serial No. 09/761,904, filed January 17, 2001 entitled "END-TO-END RESPONSE TIME MEASUREMENT FOR COMPUTER PROGRAMS USING STARTING AND ENDING QUEUES", by Paul F. Klein et al., attorney's docket number 30695.12-US-C2, which application is a continuation of United States Patent No. 6,202,036, issued on March 13, 2001, Application Serial No 09/428,271, filed October 27, 1999 entitled "END-TO-END RESPONSE TIME MEASUREMENT FOR COMPUTER PROGRAMS USING STARTING AND ENDING QUEUES", by Paul F. Klein et al., attorney's docket number 30695.12-US-C1, which application is a continuation of United States Patent No. 5,991,705, issued on November 23, 1999, Application Serial No. 08/899,195, filed July 23, 1997, entitled "END-TO-END RESPONSE TIME MEASUREMENT FOR COMPUTER PROGRAMS USING STARTING AND ENDING QUEUES," by Paul F. Klein et al., attorney's docket number 30695.12-US-01; and

United States Patent Application Serial No. 09/428,262, filed October 27, 1999 entitled "ROUND TRIP RESPONSE TIME MEASUREMENT FOR COMPUTER PROGRAMS", by Paul F. Klein et al., attorney's docket number 30695.15-US-01.

BACKGROUND OF THE INVENTION

1. Field of the Invention.

The present invention relates generally to obtaining information across a network, and more specifically to reducing the transmission time of such information by adapting the size of the information to the speed of a network connection.

2. Description of the Related Art.

In today's electronic and Internet environment, the speed used while accessing and transmitting data over the Internet is a major success factor for an Internet company attempting to conduct income-generating commerce over the Internet. Specifically, with the popularity of the world wide web and the Internet, the real-time taken to access a web site and conduct commerce with that site must be within a reasonable time-frame (e.g., under 8 seconds per web page request). Additionally, a web site engaging in income-generating commerce over the Internet may lose customers if that web site's real-time access exceeds the reasonable time-frame. Such an income loss may occur due to the fact that the customer, engaging in commerce with that web site, will become impatient with the slow access, cancel the transaction and select another competitive web site that offers the same type of commerce at a faster speed.

Most web sites supplying electronic commerce that achieve less than a reasonable time-frame for a web transaction find it difficult to achieve a reasonable time-frame. When the need for obtaining a reasonable time-frame is added to the complexity of varying Internet and network speeds feeding to a web site, produced by various numbers of Internet Service Providers (ISP's), it is almost impossible to create an electronic commerce web site with just the right amount of information that will guarantee everyone a reasonable time-frame for real-time access.

BRIEF DESCRIPTION OF THE DRAWINGS

Referring now to the drawings in which like reference numbers represent corresponding parts throughout:

FIG. 1 schematically illustrates a hardware and software environment in accordance with one or more embodiments of the invention; and

FIG. 2 is a flow chart illustrating the measuring of a speed of a network connection and adapting the retrieval of information based on the network connection speed in accordance with one or more embodiments of the invention.

SUMMARY OF THE INVENTION

Making the transmission of information faster between a client and server is very desirable for the successful implementation of electronic commerce over a network such as the Internet. One method of accelerating the information/data transmission is to reduce the amount of data being requested thus transmitted, over a network connection. The technique used for dynamically making this happen, during the course of a network connection's life, is called adaptive response time.

Adaptive techniques revolve around measuring a response time between a client and server and using the results of the measurement to determine how much information (or how large an object) can be transmitted in a reasonable amount of time. When it is determined that requested information will take too long to transmit, the client or server adapts the request for reduced information (i.e., less information or an object of lesser size) that will take less time to transmit. Additionally, if it is determined that requested information will be transmitted very quickly across a high bandwidth network connection, the client or server may adapt the request for enhanced information (i.e., more information or an object of greater size) to provide the client with enhanced/additional capabilities/information.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

In the following description, reference is made to the accompanying drawings, which form a part hereof, and in which is shown, by way of illustration, one or more embodiments of the invention. It is understood that other embodiments may be utilized and structural and functional changes may be made without departing from the scope of the present invention.

Overview

Embodiments of the invention increase the speed with which a client retrieves information and content from a server and displays such information. The speed of a network connection (i.e., the Internet connection) is measured. Thereafter, the size of the information to be retrieved across the network connection is adjusted based on the speed of the network.

Hardware Environment and Software Embodiments

FIG. 1 schematically illustrates a hardware and software environment in accordance with one or more embodiments of the invention, and more particularly, illustrates a typical distributed computer system 100 using a network/network connection 102 to connect client computers 104 to server computers 106. A typical combination of resources may include a network 102 comprising the Internet, LANs (local area networks), WANs (wide area networks), SNA (systems network architecture) networks, or the like, clients 104 that are personal computers workstations, minicomputers, etc., and servers 106 that are personal computers, workstations, minicomputers, mainframes, etc. Additionally, both client 104 and server 106 may receive input (e.g., cursor location input) and display a cursor in response to an input device such as cursor control device 108.

In accordance with one or more embodiments of the invention, network 102 (such as the Internet) connects client computers 104 executing applications such as adaptive agent 110 to server computers 106 executing applications such as adaptive selector 112. Applications such as adaptive agent 110 and adaptive selector agent 112 may be written in a suitable programming language that is applicable to the computer hardware and software environment they need to execute on. For example, such applications 110 and 112 may be written in a portable language such as Java that is conducive to components that make up an industry standard Internet environment.

In one or more embodiments of the invention, client 104, server 106, and network 102 comprise elements in the Internet environment. In such embodiments, client 104 may comprise or be equivalent to an Internet web browser such as NETSCAPE NAVIGATOR or MICROSOFT INTERNET EXPLORER, and adaptive agent 110 is equivalent to an applet (e.g., a Java applet) obtained from server 106. An applet is a program (usually small in size) that is downloaded from the server 106 and run from the browser on client 104. If the applet is written in the Java programming language, a Java virtual machine may be built into the browser and interprets the instructions.

In the Internet environment, server 106 may be a web server 106 currently available in the market such as the Web Server available from Netscape, the Internet Information Server (IIS) available from Microsoft, or the Web Server available from Apache. In addition, adaptive selector 112 is equivalent to a web server filter (e.g., the web server filter found in Microsoft's IIS Web Server). The use of the invention on the Internet and the use of the remaining components 114-120 are described in detail below.

Generally, components 110-124 all comprise logic and/or data embodied in or retrievable from a device, medium, signal, or carrier, e.g., a data storage device, a data communications device, a remote computer or device coupled to the computer across a network 102 or via another data communications device, etc. Moreover, this logic and/or data, when read, executed, and/or interpreted, results in the steps necessary to implement and/or use the present invention being performed.

Thus, embodiments of the invention may be implemented as a method, apparatus, or article of manufacture using standard programming and/or engineering techniques to produce software, firmware, hardware, or any combination thereof. The term "article of manufacture" (or alternatively, "computer program product") as used herein is intended to encompass logic and/or data accessible from any computer-readable device, carrier, or media.

Those skilled in the art will recognize many modifications may be made to this exemplary environment without departing from the scope of the present invention. For example, those skilled in the art will recognize that any combination of the above components, or any number of different components, including different logic, data, different peripherals, different software, and different devices, may be used to implement the present invention, so long as functionality as described below are performed thereby.

In accordance with one or more embodiments of the invention, software applications executing in system 100 such as the adaptive agent 110 and adaptive selector 112 provide for adapting the transmission/retrieval of information across network 102 to the speed of network connection 102's speed. Adapting the retrieval of information to a network connection 102's speed is based on removing data from the information in such a way, as to keep the meaning of the information intact while decreasing that size of the information. The adapting process uses measurements taken of the network connection 102 speed and dynamically deduces what type of information alteration is to be used prior to sending the information over the network 102. One or more embodiments of the invention combine network connection 102 measurement and information adaptation.

Information transmitted across a network may be maintained in and referred to herein as an object (also referred to as an Internet object) in an object-oriented programming environment executing on client 104 or server 106. Alternatively, for purposes of this application, an object may refer generally to a unit of information that may be transmitted, used, retrieved, or otherwise accessed on or across a network 102. Information/objects that can be defined in various sizes without losing their meaning are called adaptable information/objects. An example of adaptable information is a web graphic.

A web graphic is a picture defined as data, flowing over any given network 102 connection that contains a commercially digital encoding of a viewable art-form in one of the industry standard formats (e.g., JPEG [joint photographics expert group] or GIF [graphics interchange format]). Each graphic, as digitally encoded, may typically contain a stream of data bytes, large in size compared to plain text, which consume lots of real-time to transmit across a network 102 connection.

Varying sizes of an adaptable object may be utilized depending on the particular object in question. With a graphic object, the commercially available digital encoding (JPEG or GIF) may already provide a certain amount of data reduction from the art-forms original form. However, to reduce the size of the graphic even further and still maintain its visual meaning, the graphic can be made physically smaller visually, can have its color diminished or completely removed, or can have its color removed with reduced shades of gray. Any one of these acts substantially reduces the stream of data bytes required to transmit that graphic over the network 102 and thus reduces the amount of real-time that the transmission of the graphic will take.

Depending on the object and information being utilized and transmitted, adaptation may not be possible or useful for all objects transmitted across or utilized on a network 102, client 104, or server 106. For example, some objects must be transmitted in their entirety without change. However, most of the more utilized objects are likely adaptable.

The adaptation and transmission of an object across network 102 may be automatic, being called upon at the moment that an object is ready for transmission. Once data is removed from the object, its size is reduced and the goal of faster and more efficient network 102 transmission can be achieved.

Embodiment Details

Referring to FIG. 1, the components of system 100 provide the ability to measure the speed of a network connection 102 and adapt the size of an object based on the measurement. In one or more embodiments of the invention, client 104 issues object requests to server 106 through adaptive agent 110 and adaptive selector 112, allowing adaptive selector 112 the ability to make adaptive decisions on which object library (116, 118, or 120) the request is to be obtained from.

When client 104 makes a request of server 106 through adaptive agent 110, adaptive agent 110 may delay the request first, then issue its own network 102 request over network 102 to adaptive selector 112. Adaptive agent 110 issues a request to the calibrated object library 124 for an object/information with pre-known size and properties. Adaptive agent 110 measures the time it takes to retrieve this calibrated object by setting a software stopwatch prior to and just after the calibrated object is returned. This round-trip response time of the calibrated object, from the calibrated object library 124 is used to compute the current speed of the network connection 102 between client 104 and server 106. Alternatively, adaptive agent 110 may send a nominal request to adaptive selector 112 and measure its round trip response time similar to that described above. A nominal request is known in the computer industry as a ping.

Using the known size of the calibrated object (or based on the ping results), a bytes-per-second metric can be computed and this metric is sent from adaptive agent 110 to adaptive selector 112 across network 102. Adaptive selector 112 can utilize this value later when the adaptation processes takes place, but for now just saves it away. Adaptive agent 110 may send a new computation of bytes-per-second at any time to adaptive selector 112, and in doing so, replaces the previous computation saved away at adaptive selector 112.

Once adaptive selector 112 receives network 102 connection speed information from adaptive agent 110, adaptive agent 110 takes the original request out of delay and processes it as a standard request from client 104 and sends the request to adaptive selector 112 over network 102. Based on the current network connection 102 speed previously calculated, adaptive selector 112 adapts client 104's request, if it is an adaptable object, by selecting a replacement object (e.g., one of the three possible object library 114-118 replacements).

Objects in object library 114-118 are conceptually duplicate except that they contain less data while conveying equivalent meanings. As described above, an example of this kind of adaptable object is a graphic that can be made smaller or larger, but still retain its visual meaning. If the network connection 102 speed between client 104 and server 106 proves to be measured as very slow, below a threshold acceptable to client 104 and server 106, a smaller object may be chosen from the object library (small) 118. If the network connection 102 speed is measured to be very fast, a larger object may be chosen from object library (large) 114. Thus, client 104's request has been adapted to fit the optimal performance of the current network connection 102.

Operation over the World Wide Web

A more specific implementation, one that solves an Internet electronic commerce issue, is in guaranteeing faster access to commercial Internet web sites. In this implementation, client 104 (or an application executing on client 104) is equivalent to a commercially available Internet browser, adaptive agent 110 is equivalent to a Java applet that is obtained from Java library 122 and server 106 is equivalent to a commercially available web server. In addition, adaptive selector 112 is equivalent to a standard web server filter such as that found in Microsoft's IIS Web Server.

In this operation, web browser 110 makes a request for a web page from web server 106. That web page is retrieved from web page library 120 and intercepted by web server filter 112 where an applet tag is dynamically inserted into the web page. This tag references an applet that resides in Java library 122. The resulting web page is then sent to the web browser 104 over Internet network 102. Applet 110 executes in the web browser 104 and behaves just like an adaptive agent 110. Applet 110 makes a request to web server filter 112 for a calibrated object from the calibrated object library 124. The object is returned to web browser 104 via Internet network connection 102. Web browser 104 then times the retrieval of the calibrated object and computes a bytes-per-second response time metric by taking the total number of bytes in the calibrated object and dividing it by the number of seconds that transpired to retrieve that object. The results of that calculation are sent to web server filter 112 via Internet network connection 102 and saved away for later use. Alternatively, as described above, web browser 104 may ping the web server 106 to determine the speed of the Internet network connection 102.

Web browser 104 then resolves the web page results (e.g., the HTML [hypertext markup language] or XML [extensible markup language] as part of normal web browser 104 function) and makes additional requests for web page objects defined by the web page HTML or XML text. Accordingly, web browser 104 makes additional requests, for these resolved objects, to/from web server 106. Each object requested may be intercepted by web server filter 112 where adaptation of the object takes place. The required speed of the requested object is calculated by comparing its size to that of the calibrated object that was previously obtained from calibrated object library 124. If the requested object is calculated to take a long time to retrieve, web server filter 112 adapts the request to an object of lesser size from object library (small) 118 and retrieves that object from the small library 118.

If the requested object is calculated to be quick in speed to retrieve/transmit (e.g., if the Internet connection 102 is a high bandwidth connection), a larger more complex object could be used instead from object library (large) 114. By selecting a large object, web server filter 112 may provide the web browser 104 (i.e., the requestor) more information than a user/web browser 104 on a slower Internet connection 102. Thus, this method provides an optimal amount of information for the specific speed of a given Internet connection 102.

Variation to the Operation over the World Wide Web

One or more variations of the above-described world wide web embodiments may be implemented. In one or more variations, web browser 104 makes a request for a web page from web server 106. Instead of the web page, from web page library 120 containing the actual web object, the web page contains an applet definition in its place. As parameters to that applet, the names of the three possible adaptive objects may be provided. The web page is then served up to the web browser 104 over Internet connection 102 as normal. Web browser 104 resolves the web page, encounters the applet definitions, and then executes them as part of any commercial web browser 104 function.

Each applet 110 tests the Internet connection 102 response time between web browser 104 and web server 106 by pinging web server filter 112 and measuring its round trip response time similar to that described above. Based on the applet's 110 measurement of the current Internet connection 102 response time, the applet 110 may make a request for one of the three adaptive objects from one of the three object libraries 114-118 at web server 106. Applet 110 may (or may not) make this request directly, bypassing web server filter 112. Web server 106 passes back the specific object directly to the applet 110 that made the original request. The applet then renders the object at the web browser 104 in a standard way. In doing so, each applet 110 makes the decision to which sized object to retrieve from web server 106 that is optimal for it's current Internet connection 102. Alternatively, applet 110 may be used to determine the speed of the Internet connection 102 while the determination of the particular object to transmit remains with web server filter 112.

Program Flow

FIG. 2 is a flow chart illustrating the measuring of a speed of a network connection 102 and adapting the retrieval of information based on the network connection 102 speed in accordance with one or more embodiments of the invention. Steps 202-206 provide for the determination of a speed of a network connection 102. At step 202, a request for information of a known size is transmitted. The information of the known size is obtained at step 204 and the speed of the network 102 is determined at step 206 based on the round-trip response time of the information. Such a determination may simply comprise a ping or may involve further details and object comparisons as described above. Further, as described above, one or more of steps 202-206 may be performed by client 104 (e.g., by an adaptive agent 110 such as a web browser or by an applet) or by a server 106 (e.g., by an adaptive selector 112 such as a web server filter).

Alternatively, the speed of the network connection 102 may be determined as described in one or more of the following co-pending and commonly assigned patent applications which applications are fully incorporated by reference herein:

United States Patent Application Serial No 09/711,660, filed November 13, 2000 entitled "METHOD AND APPARATUS FOR DETERMINING A RESPONSE TIME FOR A SEGMENT IN A CLIENT/SERVER COMPUTING ENVIRONMENT", by Paul F. Klein et al., attorney's docket number 30695.19-US-U1, which application claims the benefit of Provisional Application Serial No. 60/172,026, filed December 23, 1999, by Paul F. Klein, entitled "MEASURING RESPONSE TIME FOR VARIOUS SEGMENTS OF A STANDARD CLIENT/SERVER COMPUTING ENVIRONMENT BY DIRECTLY MEASURING ONE SEGMENT AND STATISTICALLY DERIVING RESPONSE TIME FOR THE REST," attorney's docket number 30695.19-US-P1;

United States Patent Application Serial No. 09/761,904, filed January 17, 2001 entitled "END-TO-END RESPONSE TIME MEASUREMENT FOR COMPUTER PROGRAMS USING STARTING AND ENDING QUEUES", by Paul F. Klein et al., attorney's docket number 30695.12-US-C2, which application is a continuation of United States Patent No. 6,202,036, issued on March 13, 2001, Application Serial No 09/428,271, filed October 27, 1999 entitled "END-TO-END RESPONSE TIME MEASUREMENT FOR COMPUTER PROGRAMS USING STARTING AND ENDING QUEUES", by Paul F. Klein et al., attorney's docket number 30695.12-US-C1, which application is a continuation of United States Patent No. 5,991,705, issued on November 23, 1999, Application Serial No. 08/899,195, filed July 23, 1997, entitled "END-TO-END RESPONSE TIME

MEASUREMENT FOR COMPUTER PROGRAMS USING STARTING AND ENDING QUEUES," by Paul F. Klein et al., attorney's docket number 30695.12-US-01; and

United States Patent Application Serial No. 09/428,262, filed October 27, 1999 entitled "ROUND TRIP RESPONSE TIME MEASUREMENT FOR COMPUTER PROGRAMS", by Paul F. Klein et al., attorney's docket number 30695.15-US-01.

At step 208, a determination is made regarding the information to be obtained across network connection 102 based on the speed of the network. This determination may be made by the client 104 (e.g., by adaptive agent 110) or by the server 106 (e.g., by adaptive selector 112 or another application executing on server 106). The determination comprises evaluating the speed of the network connection 102 and electing to obtain information of a reduced size as the speed of the network connection 102 decreases. Thus, information may be stored in various sizes in libraries 114-118 and is obtained by client 104 depending on the speed of the network connection 102. For example, as described above, graphic information may be stored in various sizes varying from being physically smaller visually, to having diminished color, to having color removed and reduced shades of gray. At step 210, the information is obtained/retrieved across network connection 102 from server 106 to client 104.

Conclusion

This concludes the description of one or more embodiments of the invention. In summary, adaptive technology allows a server to adapt the delivery of information to optimize a specific network connection, allowing faster and more efficient transfer of those objects. By using technology that measures the network speed to a given server, a client can pre-determine the amount of real-time required to transfer information. If this real-time is unacceptable (e.g., for successful Internet commerce), the size of the information to be obtained is adaptively reduced until the real-time requirements meets or exceeds the client's desires (e.g., until it complies with successful Internet commerce standards). By pre-measuring a network connection's speed, information being transmitted can be adapted in some fashion, to the connection speed, before being sent. Accordingly, the information's transmission time is optimized.

The foregoing description of one or more embodiments of the invention has been presented for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed. Many modifications and variations are possible in light of the above teaching. It is intended that the scope of the invention be limited not by this detailed description, but rather by the claims appended hereto.